IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

IMPLICIT, LLC, Plaintiff,	§ § §	
v.	§	Case No. 2:19-cv-00040-JRG-RSP
IMPERVA, INC., Defendant.	§ § §	

PLAINTIFF IMPLICIT, LLC'S REPLY CLAIM CONSTRUCTION BRIEF

Pursuant to Local Patent Rule 4-5(c), Plaintiff Implicit, LLC respectfully submits this Reply Claim Construction Brief.[1]

I. Collateral Estoppel Does Not Apply

Collateral estoppel does not apply here. First, Imperva concedes that the present issues and those resolved in the *NetScout* Claim Construction Opinion are not identical—as it characterizes the issues as being only "*substantially* identical." Resp., at 7 (emphasis added). But this is not the standard. The *NetScout* Opinion resolved disputes concerning twenty-four (24) differently-worded claim terms. 2:18-CV-53 (Dkt. No. 111). Thirteen (13) of these claim terms are listed in the *Sandvine* Stipulation that resulted in the judgment in that case. 2:18-CV-54 (Dkt. Nos. 17 & 18). Here, the issues are different. The parties dispute claim terms that were not construed in *NetScout* (the "one or more routines" term and the "session" terms) and three of the patents-in-suit in this action (the '780 Patent, '839 Patent, and '378 Patent) were not asserted in the *NetScout* and *Sandvine* cases. Further, issues original to this action include: (i) whether—as Implicit argues—the Court should reject Imperva's "source code is the path" defense for the "sequence of routines" terms; and (ii) Imperva's position that a pointer cannot be used to identify an "outermost header" as that term was used in the Court's prior constructions of the "execute" and "convert" terms. *See* Dkt. No. 208 7-14, 17-27. These differences are "material" and, therefore, foreclose the application of collateral estoppel here. *See Va. Innovation Scis., Inc. v. Amazon.com, Inc.*, Civil Action No. 4:18-CV-474, 2019 U.S. Dist. LEXIS 152664, at *100 (E.D. Tex. Sep. 9, 2019) (collateral estoppel did not bar construing a term despite the fact that it had

[1] Implicit submits this Reply Claim Construction Brief in its case against Imperva. The Court has stayed Implicit's proceedings against Fortinet and Juniper. Dkt. No. 230. Implicit reserves its right to file a Reply in those cases should the Court lift the stay.

been previously construed in a related patent because the "context" of the surrounding claim language created a "material difference" between the terms).

Second, the previously construed terms were not essential to the judgments in either the *NetScout* case or the *Sandvine* case. "[W]hen 'alternative grounds' for relief exist, none of the grounds are strictly 'necessary to the judgment,'" and, therefore, cannot serve as a basis for invoking collateral estoppel. *Blitzsafe Tex., LLC v. Honda Motor Co.*, 2016 Dist. LEXIS 123572, at *24 (E.D. Tex. Sep. 12, 2016) (quoting *Hicks v. Quaker Oats Co.*, 662 F.2d 1158, 1168-69 (5th Cir. 1981)). The *NetScout* trial ended with a general verdict of noninfringement and, given that NetScout asserted multiple infringement defenses, it is indeterminable from the record which ground(s) was essential to the jury's determination. Similarly, the *Sandvine* Stipulation only provides that noninfringement exists based on certain applicable constructions, i.e., a determination that is materially identical to the one made by a jury when it is charged to apply certain claims constructions and finds noninfringement. In both situations, each construction issue is only a possible ground for the judgment. *Allergan Sales* is consistent with the foregoing conclusions. In that case, this Court conducted a *bench* trial and made specific findings of *infringement*. *Allergan Sales, LLC v. Sandoz Inc.*, No. 2:12-cv-207-JRG, 2016 U.S. Dist. LEXIS 41013, at *22 (E.D. Tex. Mar. 29, 2016). In such a situation, the conclusion is logically compelled that the applicable claim constructions were "essential" to the judgment, because each asserted claim limitation *must* be present in the accused products to support a finding of infringement. The converse is true here—of the multiple claim construction issues, it is indeterminable which one(s) acted as the "essential" ground for the jury's finding of noninfringement given that NetScout's noninfringement defense raised multiple issues.

And third, the *NetScout* and *Sandvine* decisions have not been fully litigated and therefore it would be unfair to apply preclusion here. The *NetScout* case is still pending before this Court on post-trial motions, and the Federal Circuit just recently docketed the *Sandvine* appeal.

II. "Sequence of [Two or More] Routines" / "List of Conversion Routines"

Imperva's proposed construction will confuse the jury by allowing Imperva to incorrectly argue noninfringement on the basis that its "chain of modules"—a term not used in the Patents—is the source code, and that the source code connected these modules before the device is put into operation and before the first packet of a message arrives. That is not what that claim term means. *See* 2:18-CV-53, Dkt. No. 111, at 13 ("The patentee did not disclaim the *existence* of software routines prior to receiving a first packet of the message."). That argument is not proper, and the Court should foreclose it here.

The source of confusion flows from how Mosberger uses "chains" of "modules" versus how Imperva will use its construction at trial. The terms "chains" and "modules" are not included in the claim language and are not terms used by Implicit; not used in the intrinsic evidence of the patents-in-suit to describe the claims; or used to describe the invention. They are terms from Mosberger. Mosberger discloses creating a "module graph" structure, which "consists of a collection of modules whose services are connected in a (hopefully) meaningful manner." Dkt. No. 228-1, at 11 (quoting Mosberger, at 65). That module graph is "configured at build time," which is when the software is built. *Id.* at 12 (quoting Mosberger, at 71). The module graph is a structure—it is not a flow chart showing how the system may process packets, and it is not the source code itself. The Accused Products do not use a "module graph" or anything analogous.

The module graph is also not the "path" in Mosberger. The reference discloses creating the "path"—shown in the green line below—*prior to* receiving the first packet of a message:

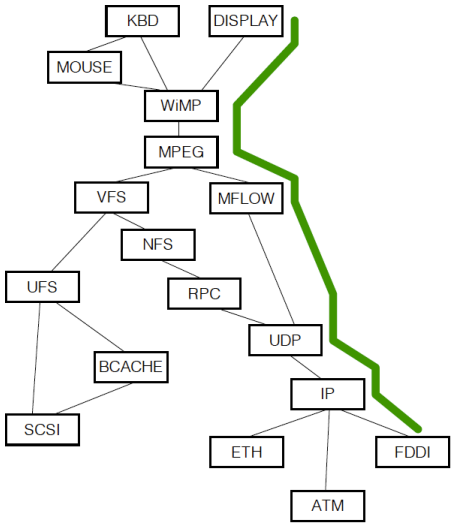

Figure 2.4: Example Path in Modular System

In the Mosberger figure above, the boxes connected by the gray lines reflect the "module graph." The green line reflects the "path." That "path" is a data structure (the "path object"). Mosberger explains that its path-creation functionality "creates the actual path object, inserts the stages into it, and ***establishes the various chains through the path structure***." Dkt. No. 228-1, at 11 (quoting Mosberger, at 81) (emphasis added). Thus, according to Mosberger, the "chains" of modules are "established" in the path structure when it is created. There is no analogous pre-created "path" data structure in the Accused Products.

Imperva's construction erases the concept of a pre-created path (or pre-created module graph) from this discussion of Mosberger. But that distinction is critical to the discussion of Mosberger. There are no module graphs or pre-created path data structures in the Accused Products. And Imperva will use its proposed construction to erase the "path" concept from Implicit's claims and inject a different concept: that the source code "modules" in Imperva's Accused Products are "connected" via the source code in ways that are established before the first packet arrives. That is how virtually all software operates. And Implicit did not disclaim that concept, as prior courts have held: "Implicit did not disclaim the ability to create a sequence of

conversion routines by relying in some part on predefined 'configuration information,' but only the use of pre-configured paths." *F5 Networks I*, at 6.

Implicit's proposed construction captures this concept by focusing on whether the arrangement was pre-configured, *viz.*, identified in a path structure before receiving the first packet of a message. Imperva's proposed construction omits the notion of pre-created path structures. It instead provides a confusing negative limitation that Imperva will use to focus only on the routines and how they might be arranged. Imperva's proposed construction cannot be reconciled with the claim limitation requiring that the path be created "based on an identification of information in a received packet of a message."

III. "One or More Routines"

This term was not at-issue in *NetScout*—the Patents that contain this term were not asserted in that case—and it presents a different issue than the "sequence of [two or more] routines" terms. This fact is important for two reasons: (1) it shows why collateral estoppel cannot apply for these terms (see above), and (2) it shows that Imperva's construction is incorrect on the merits.

During prosecution of the patents that contain the "one or more routines" terms, Implicit expressly rescinded any disclaimer made during prosecution of Implicit's prior patent applications that formed the basis of the prior constructions of "sequence of [two or more] routines." *See e.g.*, Dkt. No. 208-3, -4, and -5 (rescinding disclaimer); *Hakim v. Cannon Avent Grp., PLC*, 479 F.3d 1313, 1318 (Fed. Cir. 2007) (reciting standard for rescinding a prosecution history disclaimer). Implicit expressly stated that it is "explicitly seeking to avoid a construction of the claims adopted by the F5 court, in which a 'sequence' cannot refer to 'pre-identified' sequences of routines. To this end, Applicant has omitted the phrase 'sequence of two or more routines' from the claims." Dkt. No. 208-3, at 8. Imperva's proposed construction is incorrect because it expressly excludes from the scope of these claims the use of pre-identified sequences.

Imperva wholly ignores that file history. It instead seeks to cherry-pick a single sentence that states that only certain statements from the June 6, 2013, preliminary amendment during the prosecution of the '683 Patent were incorporated by reference. Resp., at 16. But, when read fully, the entire section of that amendment is directed to withdrawing the disclaimer. And the actual pages incorporated do not disclaim the use of pre-identified sequences, reflected in the Court's primarily reliance on statements from the file history of the parent of the '163 Patent in *NetScout*. *See* 2:18-CV-53, Dkt. No. 111, at 11–13. Those pages instead explain that the Mosberger reference does not disclose creating paths after receiving the first packet of a message. *See* 2:18-CV-53, Dkt. No. 111, at 13 (quoting June 6, 2013 Preliminary Amendment, at 11 & 12). That is already expressly required by the claim language of the '683 Patent, which requires creating a path *after* receiving the first packet of a message. *E.g.*, '683 Patent, claim 1 ("create, based on an identification of information in a received packet of a message, a path that includes one or more data structures that indicate a sequence of routines for processing packets in the message"). There is no disclaimer—especially no disclaimer that carries into the "one or more routines" terms given these Patents' unique file history.

These terms should instead be given their plain-and-ordinary meaning. Substantial guidance is provided by the claims themselves. Claim 1 of the '780 Patent, for example, recites "identifying, using the key value, one or more routines for processing the packet." '780 Patent, claim 1. Imperva's construction does not explain what a "routine" is or what "routines for processing the packet" are. It instead includes confusing negative limitations and new concepts—"[two or more] software routines arranged in a sequence that was not established in a chain of modules connected before receiving a first packet of the message"—that do not find support in the intrinsic evidence of the Patents and come from prosecution history that these Patents expressly

rescinded. Imperva's construction will only assist Imperva in raising a confusing and improper "the source code is the path" noninfringement argument at trial. It should be rejected.

IV. The "Execute a Transmission Control Protocol (TCP) to Convert One or More Packets Having a TCP Format Into a Different Format" Terms

The term "outermost header" should not be imported in the construction of an otherwise understandable term, "format." The specification does not contain the term "outermost header," nor is it depicted in the figures. Despite that, Imperva incorrectly asserts that the written description "plainly demonstrates that the 'format' of a packet is its 'outermost header structure.'" Resp., at 21 (citing specification). Indeed, the claims themselves indicate that the "format" of a packet is not necessarily determined by its "outermost header" because some claims expressly refer to an "outermost header" and some do not. *Compare, e.g.*, '683 Patent, claim 16 (reciting a routine that is used to execute a protocol to "to convert packets from an input format to an output format") *with* claim 20 (reciting the apparatus of claim 16 wherein "the particular routine is executable to convert packets by removing an outermost header of the packets"). Even Imperva will admit that the packets flowing through the Accused Devices are Ethernet-IP-*and*-TCP-formatted packets. That is the very nature of the TCP/IP protocol stack.

Imperva's construction is sourced from the file history. But Imperva does not argue prosecution history disclaimer—a high bar that is not met here. *See, e.g.*, *Thorner v. Sony Comp. Ent't Am., LLC*, 669 F.3d 1362, 1366–67 (Fed. Cir. 2012) ("To constitute disclaimer, there must be a clear and unmistakable disclaimer."). And Imperva does not address the entirety of the prosecution history, which explains how the Implicit Patents process packets one layer at a time, *viz.*, starting at the Ethernet layer, then the IP layer, then the TCP layer. Dkt. No. 228-2, 6–7. Because the prosecution history undisputedly lacks the disclaimer required to limit the ordinary meaning of the term "format," Imperva's proposed construction should be rejected.

Imperva's response brief shows that it seeks to use the "outermost header" term to exclude from the scope of the claims any system that uses pointers to access a packet's encapsulated headers. Imperva's position would exclude virtually every real-world computer system, including any systems written in a C-based language, any systems based on BSD, and any system based on Linux. That position is driven purely by Imperva's noninfringement theory (most network devices are built on Linux-type or BSD-type processing). And Imperva fails to appreciate that the Court denied summary judgment on this pointer-based defense in the *NetScout* case. 2:18-CV-53 (Dkt. No. 206), at 60:22–61:4. Implicit, accordingly, requests that the Court hold that creating a pointer to a header in a routine can define the format of the packet being processed in that routine.

Imperva also does not substantively engage with any of the intrinsic evidence cited on the face of the Implicit Patents—all of which undisputedly shows that systems (especially C-based software) have long used pointers to traverse the protocol stack. *See* Opening Br., at 20–27 (collecting references); *see also* Dkt. No. 208-1, at ¶¶ 109–156. Nor does Imperva address any of the real-world systems that use pointers to traverse the protocol stack (*e.g.*, Linux). *See id.* Imperva instead takes the extreme position that the claims do not cover ***any*** type of pointer-based system due to Implicit's usage of the term "outermost header" in the file history when discussing the Decasper router. *See id.* But nothing in the file history or the phrase "outermost header" excludes the use of pointers. Decasper itself shows that the term "outermost header" does not exclude pointer-based systems because the "outermost header [that] is always IP" in Decasper is defined by a pointer (the source code `struct ipv6 *ip`). Dkt. No. 208-6, at ¶ 6; Dkt. No. 208-10, at DEFSPA182329; *see also* Dkt. No. 208-1, at ¶¶ 125–35. Implicit distinguished Decasper during prosecution on the basis that it does not disclose inspecting the packets at the TCP level and beyond (it only had an "outermost header" of IP)—not on the basis that Implicit's Patents

excluded the use of pointers to traverse the protocol stack. Again, that would exclude virtually all systems.

Imperva also relies on the Court's construction in *Palo Alto Networks* in which the Court concluded that "removing" a header (in claim 24 of the '683 Patent) can be read as involving "modifying packets rather than merely moving a reference." Resp., at 22–23 (quoting 6:17-CV-183, Dkt. No. 101, at 26). But the Court's conclusion actually supports Implicit's position. The Court reached that conclusion because independent claim 16 did not recite the term "removal"—it only recited "convert[ing] packets from an input format to an output format," whereas dependent claim 20 expressly limited the claims to one type of conversion: "removing an outermost header of the packets." 6:17-CV-183, Dkt. No. 101, at 26. With this context, the Court concluded that "removing" a header is not the only way to convert packets up the protocol stack. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005) (en banc) ("Differences among claims can also be a useful guide in understanding the meaning of particular claim terms.").

Imperva's restrictions have nothing to do with the inventions of the Implicit Patents. One way to navigate to the TCP and application layers is to create new pointers to the different packet headers. That was the theory that the Court concluded raised a triable issue in *NetScout*. Another way is to advance a reference from one header to the next as a reference to the message is passed from one conversion routine to the next, which is disclosed in the Implicit Patents. '683 Patent, at 14:10–14 (". . . advance the reference past the header information for the protocol so that the reference is positioned at the next header."). And another way is to copy or move the packet data from one structure to another as the packet is processed up the protocol stack. Implicit's claims cover at least each of those. And how the software is programmed to navigate to the TCP and application layers is a source-code implementation detail; it is not a claim limitation.

V. The "Session Associated With a [Transport Layer/Different] Protocol" Terms

Here, Imperva ignores the claim language. Imperva does not contest that the plain-and-ordinary meaning of this term generally means a temporary instance of information exchanged between two devices. *See* Opening Br., at 27. The terms relate to a "session" that is "associated with ***a [transport layer/different] protocol***." Imperva dismisses that language and reads into these terms the construction of a different term: "state information associated ***with the message***." The "session" terms—by their very language—are associated with a ***protocol***, not a ***message***.

Indeed, even the evidence Imperva points to as "equat[ing] 'sessions' with 'state information,'" Resp., at 28, shows that Imperva is wrong: claims 4 and 10 generally recite "sessions" ***without any state information language***, whereas claims 5 and 11 add the further limitations that the "sessions ***specify state information*** for one or more of the sequence of routines, and ***wherein the state information is specific to the message***." '683 Patent, claims 4, 5, 10, and 11 (emphases added); Resp., at 28 (relying on '683 Patent, claims 5 and 11). This difference in claim language shows that "sessions" and "state information" are not necessarily the same thing in the claims—and that it would be improper to read a "state information" limitation into each "session" term. *Phillips*, 415 F.3d at 1314. Similarly, the specification and reexamination file history that Imperva relies on relating to a system that "maintains state information as an instance or sessions of conversion routine" Resp., at 28 (quoting '683 Patent, at 3:2–7) is expressly claimed in claims such as claims 5 and 11 of the ''683 Patent. It should not be read into ***every*** claim that recites a "session" associated with the ***protocol***—and not necessarily the ***message***.

VI. Conclusion

For the foregoing reasons, Implicit respectfully requests that the Court enter an order adopting Implicit's proposed constructions and rejecting the positions advanced by Defendants in their constructions.

Dated: February 25, 2020

Respectfully submitted,

By: /s/ William E. Davis, III
William E. Davis, III, *Lead Attorney*
Texas State Bar No. 24047416
bdavis@davisfirm.com
Christian J. Hurt
Texas State Bar No. 24059987
churt@davisfirm.com
Edward Chin (Of Counsel)
Texas State Bar No. 50511688
echin@davisfirm.com
Debra Coleman (Of Counsel)
Texas State Bar No. 24059595
dcoleman@bdavisfirm.com
Ty Wilson
Texas State Bar No. 24106583
twilson@davisfirm.com
The Davis Firm, PC
213 N. Fredonia Street, Suite 230
Longview, Texas 75601
Telephone: (903) 230-9090
Facsimile: (903) 230-9661

Barry Golob
bgolob@cozen.com
Thomas Fisher
tfisher@cozen.com
Cozen O'Connor
1200 19th Street, NW
Suite 300
Washington, DC 20036
Telephone: (202) 912.4815
Facsimile: (202) 618.4843

Counsel for Plaintiff
Implicit, LLC

CERTIFICATE OF SERVICE

The undersigned certifies that the foregoing document is being filed electronically in compliance with Local Rule CV-5(a). As such, this document is being served on all counsel who are deemed to have consented to electronic service. Local Rule CV-5(a)(3)(V). Pursuant to

Federal Rule of Civil Procedure 5(d) and Local Rule CV-5(d) and (e), any counsel of record not deemed to have consented to electronic service will be served with a true and correct copy of the foregoing by email on this 25th day of February, 2020.

/s/ William E. Davis, III
William E. Davis, III